

AVANCE BACK-END

Como anteriormente se mencionó, el back end esta siendo desarrollado por el Framework Laravel en el lenguaje de programación PHP.

Para este avance se levantó la Restful api, se desarrolló 3 funcionalidades y se testeo cada una en Postman.

La base de datos relacional se aloja en Google Cloud para favorecer el trabajo cooperativo y para que la app tenga más versatilidad. Para trabajar con la base de datos se implementó una estructura ORM, dicha estructura basada en Modelos, es decir cada tabla tendrá un modelo para interactuar con la misma

Funcionalidad 1: Login

Login correcto

The image displays two screenshots related to a database and API testing. The top screenshot is from MySQL Workbench, showing the 'usuarios' table in the 'mechany08' database. The table contains five rows of user data. The bottom screenshot is from Postman, showing a PUT request to 'localhost:8000/api/login' with a JSON body containing 'id' and 'password'. The response is 'login correcto'.

MySQL Workbench - usuarios Table Data:

| id | password | created_at | updated_at |
|----------|----------|---------------------|---------------------|
| usuario1 | aaa | 2020-09-01 03:47:02 | 2020-09-01 03:47:02 |
| usuario2 | aaa | 2020-09-01 03:49:05 | 2020-09-01 03:49:05 |
| usuario3 | aaa | 2020-09-01 03:51:56 | 2020-09-01 03:51:56 |
| usuario4 | aaa | 2020-09-01 03:51:56 | 2020-09-01 03:51:56 |
| usuario5 | aaa | 2020-09-01 03:52:45 | 2020-09-01 03:52:45 |

Postman - PUT Request:

URL: localhost:8000/api/login

Body (JSON):

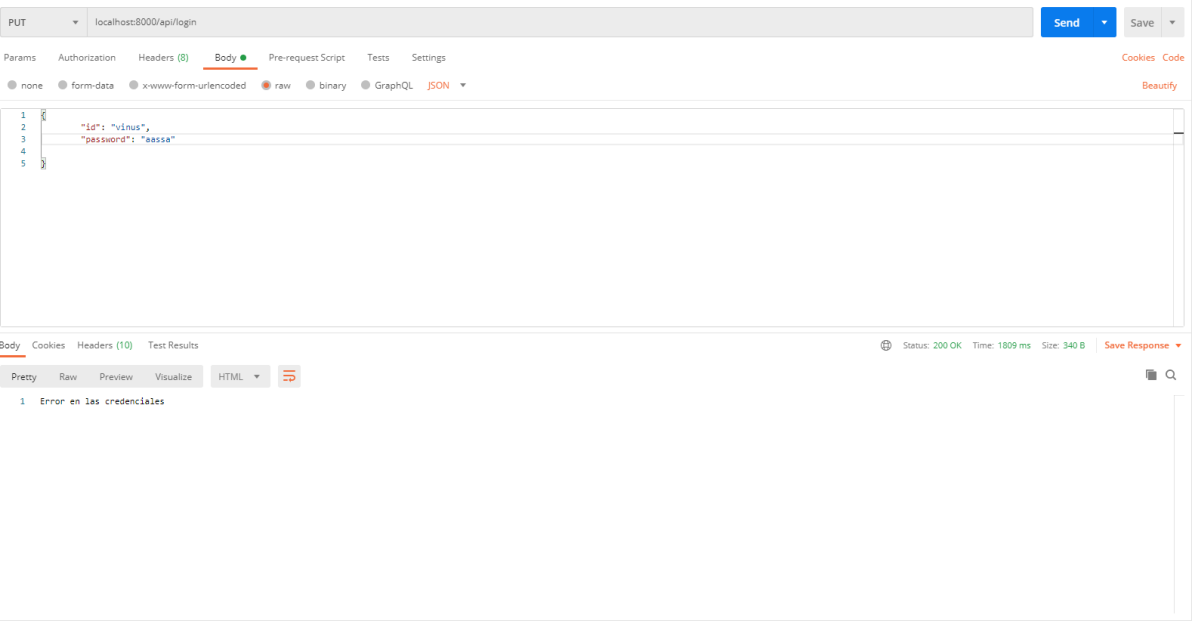
```
{  "id": "usuario1",  "password": "aaa"}
```

Response:

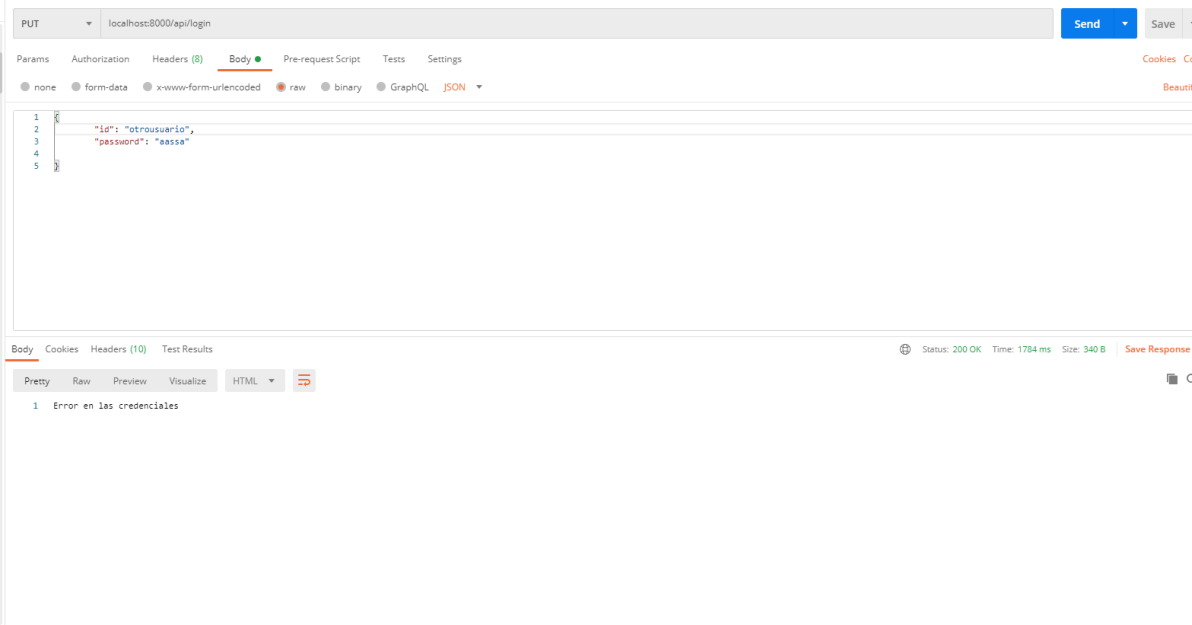
```
login correcto
```

Login incorrecto

Ingresando mal la contraseña



Ingresando mal el usuario



Funcionalidad 2: registro de usuario

POSTlocalhost:8000/api/registrou

SendSave

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1{
2  "id": "edwin",
3  "password": "hola"
4}
5
```

BodyCookiesHeaders (10)Test Results

Status: 201 CreatedTime: 1852 msSize: 430 BSave Response

PrettyRawPreviewVisualizeJSON

```
1{
2  "id": "edwin",
3  "password": "hola",
4  "updated_at": "2020-09-01T05:04:03.000000Z",
5  "created_at": "2020-09-01T05:04:03.000000Z"
6}
```

Navigator

mechanyDB

articlescategoriesclientesmedicaditemsmigrationspersonal_mecanicosventaspromocionespersonaspersonal_mecanicosusersusuariosViewsStored ProceduresFunctions

sys

ios - Tablearticles - Tableusuarios - Tableusuariosarticlesusuariospersonal_mecanicos - Tableusuarios - Tablepersonal_mecanicos - Tablepersonal_mecanicospersonal_mecanicos - Tableusuarios

1 • SELECT * FROM mechanyDB.usuarios;

Result Grid

Filter Rows

idpasswordcreated_atupdated_at

edwinhola2020-09-01 05:04:032020-09-01 05:04:03

vinusaaa2020-09-01 03:47:022020-09-01 03:47:02

vinus2aaa2020-09-01 03:49:052020-09-01 03:49:05

vinus3aaa2020-09-01 03:51:162020-09-01 03:51:16

vinus4aaa2020-09-01 03:51:162020-09-01 03:51:16

Funcionalidad 3: buscar mecánico por id

1 • SELECT * FROM mechainyDB.personal_mecanicos;

| id | categoria | calificacion | comentario | nombre | created_at | updated_at |
|----|-----------|--------------|---------------------|--------|---------------------|---------------------|
| 1 | Ruedas | 5 | Es un buen mecanico | carlos | 2023-07-01 12:00:00 | 2023-07-01 12:00:00 |

GETlocalhost:8000/api/mecanico/1

SendSave

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencoderauwbinaryGraphQLJSON

```
1{"id": "edwin",
2  "categoria": "Ruedas",
3  "calificacion": 5,
4  "comentario": "Es un buen mecanico",
5  "nombre": "carlos",
6  "created_at": null,
7  "updated_at": null}
```

BodyCookiesHeaders (10)Test Results

123456789

123456789

```
"id": 1,
"categoria": "Ruedas",
"calificacion": 5,
"comentario": "Es un buen mecanico",
"nombre": "carlos",
"created_at": null,
"updated_at": null
```

Status: 200 OKTime: 1785 msSize: 442 BSave Response